REMARKS

Claims 1-50 are pending. Claims 1-21, 25, 26 and 43-49 were rejected under 35 U.S.C. § 102(e) based on U.S. Patent No. 6,363,495 to MacKenzie. Claims 27-42 and 50 were rejected under 35 U.S.C. § 103(a) based on MacKenzie, U.S. Patent No. 6,446,219 to Slaughter and/or U.S. Patent No. 5,315,657 to Abadi. These rejections are traversed. Applicants note with appreciation that Claims 22-24 would be allowable if rewritten in independent form. For the reasons set forth below, it is believed that all claims are now in condition for allowance.

Declaration under 37 C.F.R. § 1.131

On June 2, 2003, the Applicants submitted a Declaration of Mr. Richard Frank under 37 C.F.R. § 1.131 to swear behind the MacKenzie reference. Mr. Frank's Declaration and its accompanying Exhibits (A-E), show that an example implementation of the present invention was conceived and reduced to practice prior to January 19, 1999, the effective date of the MacKenzie reference. In Exhibits A-E, for instance, several routines (e.g. objects) are presented. These include InitCm (Exhibit A) CreateCmThread, (Exhibit B) CmStartUp, (Exhibit C), InitControlDisk (Exhibit D) and WaitForCM (Exhibit E).

According to the outstanding Office Action, however, the Examiner determined that Mr. Frank's Declaration was considered ineffective to overcome the MacKenzie reference because the following claim limitations were not supported by the Exhibits:

A. Defining a shareable storage to store data for a network;

B. Granting membership in a network cluster;

C. Revoking membership of a node in a network cluster;

D. Ceasing operation of the network cluster;

E. Manager mechanism to grant membership;

F. Use of message location to grant membership; and

H. Denying membership in computer network cluster.

For the convenience of the Examiner, the Applicants will discuss below how each of the above-listed claim limitations are supported by the Exhibits in Mr. Frank's § 1.131 Declaration.

A. Defining a Shareable Storage to Store Data for a Network

Mr. Frank's Declaration and accompanying Exhibits describe an example implementation of the claimed *defining a shareable storage to store data for a network.*[1] In the example described by Mr. Frank, each node executes a cluster manager, which defines a shareable storage by writing to the shareable storage, and thus, storing data for the network. For example, the CmStartUp object discussed in Exhibit C and the InitControlDisk object discussed in Exhibit D are responsible for, among other things, defining the control area of the bound set of drives (shareable storage) to store data for a network (write to control area). The CmStartUp thread starts-up the network connection layer and initializes the shareable storage (control area in the bound set). The shareable storage that stores data for a network will be successfully defined unless the system detects an error in its initialization. The following is a discussion of some of errors that may be detected.

The definition of the shareable storage is considered unsuccessful when the CmStartUp thread determines that the shareable storage (bound set) is disabled or when it cannot be found. For instance, below are excerpts from the CmStartUp routine. As discussed below, if the control disk cannot successfully initialize, the cluster manager process executing on the node, which is attempting to initialize the shared resource, is aborted.

```
        //
        // Initialize the control area in the bound set
. . .
        LogMsg("Initializing the Control Disks");
        if (0 == pClub->CmVars.ControlDisksDisabled) {
                status = InitControlDisk(pClub);
                if (status != NEDC_IO_SUCCESS) {
                        pClub->CmFlags.CmAbort = TRUE;
                        return status;
                }
```

Further, the cluster manager software on the node checks whether the shareable storage can be used by the node to *store data for a network.* More particularly, the CmStartUp thread

[1] *See* Declaration of Richard Frank at pgs. 2-3, section 6, paragraphs 2-4; Exhibits C and D.

(Exhibit C) spawns the InitControlDisk thread (Exhibit D) to determine whether the control disk has been successfully written to. For example, it determines whether the node can write to its message location on the shareable storage. This information is stored as "status." If the InitControlDisk thread is unable to write to the shareable storage (status !=NEDC_IO_SUCCESS), then the shareable storage has not been successfully defined (Could not write Control Disk). As shown in an excerpt of the InitControlDisk code below, to successfully define the shareable storage, the cluster manager executing on the node needs the ability to perform input/output (read/write) operations on the shared disk.

```
if ((status != NEDC_IO_SUCCESS) || (UserIoReq.Status !=
NEDC_IO_SUCCESS)) {
        CmDumpToLog(CM_K_FACILITY, "Could not write Control Disk: error
        is %d\n",UserIoReq.Status);
        return UserIoReq.Status;
}
```

If the InitControlDisk thread returns the status of NEDC_IO_SUCCESS to the CmStartUp thread (Exhibit C), the cluster manager process executing on the node is aborted (pClub->CmFlags.CmAbort = TRUE).

Thus, as discussed in Exhibits C and D, when there are no errors in the initialization of the shareable storage (e.g., the node could store data at the shareable storage) the shareable storage is successfully defined (initialize control region). As such, the Exhibits of Mr. Frank's Declaration support the claimed *defining a shareable storage to store data for a network.* Therefore, the Applicants respectfully request that the rejection of independent Claims 1, 4, 5, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 32, 33, 40, 43 and 50, and their respective dependent claims based on Mackenzie be withdrawn.

B. Granting Membership in a Network Cluster

Mr. Frank's Declaration and its accompanying Exhibits describe an example implementation of the claimed *granting membership in a network cluster if the node has access to the shareable storage.*[2] For example, a node requesting membership in the network cluster executes its cluster manager, which calls the routine InitCm (Exhibit A). InitCm spawns CreateCmThread (Exhibit B) to determine the node's status, i.e. whether the node can access the shareable storage and thus be accepted as a member in the cluster (Club). Excerpts from the InitCm code (Exhibit A) are shown below.

```
/* InitializeCm creates the CM thread, and then delays return to its caller, until the
CM thread signals that */
/* it is ready to accept application registration with the CM.          */

int InitCm(short *NodeId, LPCLUSTERDB pClusterDb, NEDC_BOUND_SET
*lpBoundSet,
        PDISKDEV pDiskList, PLOGNAME pNameList)
. . .
if (!(status = CreateCmThread(pClub))) return status; /* Cm thread create failed */

LogMsg("Waiting for Cm to start");

if (!WaitForCm(pClub)) return FALSE;  // wait for Cm to initialize
```

As shown above, InitCm initializes the cluster manager on the node and spawns a cluster manager thread by running the routine CreateCmThread (Exhibit B). Then, the routine CreateCMThread spawns the CmStartUp thread (Exhibit C) to define the shareable storage. This includes initializing the disk-based messaging area (control area) on the shareable storage device (control disk). To successfully initialize the shareable storage, the node needs to write to its message location and read the cluster definition. CmStartUp invokes the InitControlDisk thread (Exhibit D) to determine whether the node can access the shareable storage and InitControlDisk returns the results as an integer (status). If the node is unable to access the repository because it

[2] *See* Declaration of Richard Frank at pgs. 2-3, section 6, paragraphs 1-5; Exhibits A-D.

does not meet the conditions ((status != NEDC_IO_SUCCESS) || (UserIOReq.Status != NEDC_IO_SUCCESS)), the node's status is returned to CmStartUp (Exhibit C), which aborts the cluster manager process by setting the pClub pointer to CmAbort = TRUE. This information is processed by the InitCm thread (Exhibit A), which then determines that the application for registration in the cluster has failed. If, however, the node successfully initializes the shareable storage and thus writes to its message location, the InitCm thread (Exhibit A) grants membership in the cluster (ready to accept application registration).

As such, the notion of *granting membership in a network cluster if the node has access to the shareable storage* is supported by Mr. Frank's Declaration. Therefore, the Applicants respectfully request that the rejection of Claims 1, 4, 5, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 32, 33, 40, 43 and 50, and their respective dependent claims based on Mackenzie be withdrawn.

C. Revoking Membership of a Node in a Network Cluster

The claimed concept of *ceasing operation of the network cluster if no node has access to the shareable storage device* is set forth in dependent Claims 3, 10, 15, 36, 42 and 49. Because the base claims are in condition for allowance, the dependent claims should also be allowed. Reconsideration of the rejection of dependent Claims 3, 10, 15, 36, 42 and 49 based on MacKenzie is respectfully requested.

D. Ceasing Operation of the Network Cluster

The claimed concept of *revoking membership of the node in the network cluster if the node ceases to have access to the shareable storage device* is set forth in dependent Claims 2, 35, 9, 14 and 45. Because the base claims are in condition for allowance, the dependent claims should also be allowed. Reconsideration of the rejection of dependent Claims 2, 35, 9, 14 and 45 based on MacKenzie is respectfully requested.

E. Manager Mechanism to Grant Membership

Mr. Frank's Declaration and accompanying Exhibits describe an example implementation of the claimed concept of *a manager mechanism to grant membership in the network cluster to the node.*[3] As discussed in sections A and B above, the software code shown in Exhibits A-E of Mr. Frank's Declaration are part of the cluster manager (CM) software code, which is compiled and executed on the node. The cluster manager software objects described in Exhibits A-E communicate with one another to determine whether the node can access the shareable repository and thus be granted membership in the cluster. For illustrative purposes, an excerpt from the InitCm code (Exhibit A) is shown below.

```
/* InitializeCm creates the CM thread, and then delays return to its caller, until the
CM thread signals that */
/* it is ready to accept application registration with the CM.          */
. . .
if (!(status = CreateCmThread(pClub))) return status; /* Cm thread create failed */
. . .
if (!WaitForCm(pClub)) return FALSE;  // wait for Cm to initialize
. . .
return  TRUE; /* ready to accept application registration */
```

Thus, the cluster manager (CM) accepts the node's application for membership in the network cluster when it receives notification from CreateCmThread (Exhibit B) and WaitForCm (Exhibit E) that the node has successfully defined and thus accessed the shareable storage. As such, the notion of a *cluster manager to grant membership in the network cluster* is supported by Mr. Frank's Declaration. Therefore, the Applicants respectfully request that the rejection of Claims 4, 20, 29, 30 and 43 and their respective dependent claims based on Mackenzie be withdrawn.

---

[3] *See* Declaration of Richard Frank at pgs. 2-3, section 6, paragraphs 1-5; Exhibits A-E.

F. Use of Message Location to Grant Membership

Mr. Frank's Declaration and accompanying Exhibits describe an example implementation of the claimed *granting membership in a network cluster to a node if the node has access to the shareable storage device, using the message location*, as in independent Claims 19, 20, 21 and 26.[4] As discussed in sections A and B above, Exhibits A-E of Mr. Frank's Declaration discuss that a node's membership in the cluster is predicated on the node's ability to access the shareable storage. In other words, the node needs to be able to use its message location (e.g. read/write to its control area) on the shareable storage (control disk). As such, the Applicants respectfully request that the rejection of Claims 19, 20, 21 and 26 and their respective dependent claims based on Mackenzie be withdrawn.

H. Denying Membership in Computer Network Cluster

Mr. Frank's Declaration and accompanying Exhibits describe an example implementation of the claimed *denying membership in the computer network cluster to a node if the node is unable to access the shareable storage device*. As discussed in Sections A-B above, Exhibits A-E of Mr. Frank's Declaration explain that the node can be granted membership in the cluster if the node can successfully access the shareable storage (control disk). Exhibits A, C and D discuss the situation where the node cannot successfully initialize the control disk and thus may not be denied membership in the cluster. For example, as discussed in the CmStartUp routine of Exhibit C, if the control disk cannot be successfully accessed, the cluster manager process executing on the node, which is attempting to request membership in the cluster, is aborted. In this way, the node is denied membership in the cluster. For the convenience of the Examiner, an excerpt from Exhibit C is shown below, which discusses aborting the cluster manager when the node cannot access the shared storage.

```
LogMsg("Initializing the Control Disks");
if (0 == pClub->CmVars.ControlDisksDisabled) {
        status = InitControlDisk(pClub);
        if (status != NEDC_IO_SUCCESS) {
```

[4] *See* Declaration of Richard Frank at pgs. 2-3, section 6, paragraphs 1-5; Exhibits A-E.

```
                pClub->CmFlags.CmAbort = TRUE;
                return status;
            }
        }
```

Further, as shown in Exhibit A, the InitCm thread will accept the node's application for registration in the cluster if the node satisfies the conditions of CreateCmThread (Exhibit B) and WaitForCm (Exhibit E). If the node does not satisfy these conditions, then the node's application for membership has failed. Thus, in the example implementation discussed in Mr. Frank's Declaration,[5] a node seeking membership in the cluster uses its cluster manager software to access the shareable storage. If the node is unable to access the shareable storage, the cluster manager is aborted, and thus, the node is denied membership in the cluster. As such, Mr. Frank's Declaration and its accompanying Exhibits support the notion of *denying membership in the computer network cluster to a node if the node is unable to access the shareable storage device.* Therefore, the Applicants respectfully request that the rejection of Claims 33, 40 and 50 and their respective dependent claims based on Mackenzie be withdrawn.

Accordingly, the software code referenced in Mr. Frank's Declaration shows that prior to January 19, 1999, a cluster membership management technique was implemented in which membership in the cluster is based on a node's ability to access to the shareable storage. As such, the claimed invention was conceived and reduced to practice prior to the filing date of MacKenzie, and therefore, the MacKenzie reference is not citeable as a prior art reference against the present application.

Rejection under 35 U.S.C. § 103(a)

Claims 27-32 were rejected under 35 U.S.C. § 103(a) based on MacKenzie and Slaughter and Claims 33-42 and 50 were rejected under 35 U.S.C. § 103(a) based on MacKenzie, Slaughter and Abadi. In light of Mr. Frank's Declaration and the discussion above, MacKenzie is not prior

[5] *See* Declaration of Richard Frank at pgs. 2-3, section 6, paragraph 2; Exhibits A-B.

art to the present application.

According to the Office Action, Abadi has been cited to show the claimed *denying membership in the computer in a network cluster to a node*.[6] Abadi discusses public/private key encryption for membership groups. Abadi describes that each membership group is represented by a certificate with an encryption key. The section of Abadi cited by the Office states in relevant part, "[a] storage device can refuse to furnish a certificate when requested, or deny its existent, but it cannot forge a bogus certificate. In most instances, failure to furnish a certificate would only deny membership in a group, and therefor deny access to some object."[7] Thus, Abadi is directed to authentication techniques using encryption and digital certificates and therefore is nonanalogous art.

With respect to Slaughter, it is noted that no particular sections of Slaughter have been cited against the claims. Therefore, it is requested that the rejection based on Slaughter be withdrawn.

It should also be noted that Claims 33, 40 and 50 discuss the inventive concept of denying membership if the node is unable to access the shareable storage regardless of network connectivity. In particular, the claims require:

- regardless of network connectivity, denying membership in the computer network cluster to a node if the node is unable to access the shareable storage device, as set forth in independent Claims 33, 40 and 50, respectively.

In this way, a node's membership in the network cluster is predicated on its ability to access the shareable storage and not its network connectivity to other nodes. As a result, the cluster can be formed from a single node that is able to access the shared repository. Also, unlike prior cluster

---

[6] Office Action, pg. 8.

[7] *See* Abadi, col. 9, ll. 55-62.

management systems, a quorum of nodes is not required to form a cluster. None of the references cited by the Examiner discuss this inventive concept of denying membership in cluster regardless of network connectivity, nor do they discuss denying membership in the cluster when the node is unable to access the shared resource.

Accordingly, it is respectfully requested that the rejection of Claims 33-42 and 50 under 35 U.S.C. § 103(a) be withdrawn.

Information Disclosure Statement

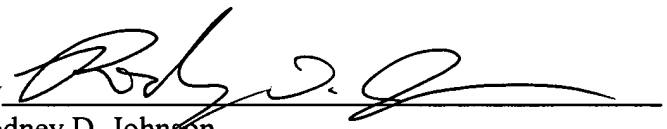
An Information Disclosure Statement (IDS) is being filed concurrently herewith. Entry of the IDS is respectfully requested.

CONCLUSION

In view of the above remarks, all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By 

Rodney D. Johnson
Registration No. 36,558
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Dated: March 8, 2004